

Convergence Analysis of the Inexact Infeasible Interior-Point Method for Linear Optimization

G. Al-Jeiroudi · J. Gondzio

Published online: 25 December 2008
© Springer Science+Business Media, LLC 2008

Abstract We present the convergence analysis of the inexact infeasible path-following (IIPF) interior-point algorithm. In this algorithm, the preconditioned conjugate gradient method is used to solve the reduced KKT system (the augmented system). The augmented system is preconditioned by using a block triangular matrix.

The KKT system is solved approximately. Therefore, it becomes necessary to study the convergence of the interior-point method for this specific inexact case. We present the convergence analysis of the inexact infeasible path-following (IIPF) algorithm, prove the global convergence of this method and provide complexity analysis.

Keywords Inexact interior-point methods · Linear programming · Preconditioned conjugate gradients · Indefinite system

1 Introduction

In this paper we are concerned with the use of primal-dual interior point method (IPM for short) to solve large-scale linear programming problems. The primal-dual method is applied to the primal-dual formulation of the linear program

$$\begin{array}{ll} \min c^T x, & \max b^T y \\ \text{s.t. } Ax = b, & \text{s.t. } A^T y + s = c, \\ x \geq 0; & y \text{ free, } s \geq 0, \end{array}$$

Communicated by Y. Zhang.

G. Al-Jeiroudi (✉) · J. Gondzio
School of Mathematics, University of Edinburgh, Mayfield Road, Edinburgh EH9 3JZ, UK
e-mail: G.Al-Jeiroudi@sms.ed.ac.uk

J. Gondzio
e-mail: J.Gondzio@ed.ac.uk
url: <http://maths.ed.ac.uk/~gondzio/>

where $A \in \mathcal{R}^{m \times n}$, $x, s, c \in \mathcal{R}^n$ and $y, b \in \mathcal{R}^m$. We assume that $m \leq n$. The primal-dual algorithm is usually faster and more reliable than the pure primal or pure dual method [1, 2]. The main computational effort of this algorithm consists in the computation of the primal-dual Newton direction of the Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} Ax - b &= 0, \\ A^T y + s - c &= 0, \\ XSe &= 0, \\ (x, s) &\geq 0, \end{aligned}$$

where $X = \text{diag}(x)$, $S = \text{diag}(s)$ and $e \in \mathcal{R}^n$ is a vector of ones.

In this paper, we focus on the use of the infeasible path-following algorithm [2, 3]. This algorithm does not require the initial point to be strictly feasible but requires only that its x and s components be strictly positive. At each iteration, the following nonlinear system needs to be solved

$$F(t) = \begin{bmatrix} Ax - b \\ A^T y + s - c \\ XSe - \sigma \mu e \end{bmatrix} = 0, \quad (1)$$

where $t = (x, y, s)$, $\mu = x^T s / n$ is the average complementarity gap and $\sigma \in (0, 1)$. We use Newton's method to solve this nonlinear system, where the direction at each iteration k is computed according to

$$F'(t^k) \Delta t^k = -F(t^k), \quad (2)$$

which yields

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = - \begin{bmatrix} Ax^k - b \\ A^T y^k + s^k - c \\ X^k S^k e - \sigma_k \mu_k e \end{bmatrix}. \quad (3)$$

Solving the linear system (3) with a direct method becomes sometimes very expensive for large problems. In these situations it is reasonable to use an iterative method. Therefore, instead of solving (2) exactly, we solve it with the inexact Newton method:

$$F'(t^k) \Delta t^k = -F(t^k) + r^k, \quad (4)$$

where r^k is the residual of the inexact Newton method. Any approximate step is accepted provided that the residual r^k is small such as

$$\|r^k\| \leq \eta_k \|F(t^k)\|, \quad (5)$$

as required by the theory [4, 5]. We refer to the term η_k as the forcing term.

In the computational practice, (3) is reduced: after substituting

$$\Delta s = -X^{-1}S\Delta x - s + \sigma\mu X^{-1}e, \quad (6)$$

in the second row, we get the following symmetric indefinite system of linear equations, usually called the augmented system:

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \quad (7)$$

where $\Theta = XS^{-1}$, $f = A^T y - c + \sigma\mu X^{-1}e$ and $g = Ax - b$.

We have dropped the iteration index k .

We use the preconditioned conjugate gradient (PCG) method to solve the augmented system (7) preconditioned by a block triangular matrix P . We have in mind a particular class of preconditioners in this paper, the ones which try to guess a “basis”, a nonsingular submatrix of A . There has been recently a growing interest in such preconditioners see for example [6–9].

In [8], the following reasoning has been used to find the preconditioner for KKT system. From the complementarity condition we know that at the optimum $x_j s_j = 0, \forall j \in \{1, 2, \dots, n\}$. Primal-dual interior point methods usually identify a strong optimal partition near the optimal solution. If at the optimal solution $x_j \rightarrow 0$ and $s_j \rightarrow \hat{s}_j > 0$, then the corresponding element $\Theta_j \rightarrow 0$. If, on the other hand, $x_j \rightarrow \hat{x}_j > 0$ and $s_j \rightarrow 0$, then the corresponding element $\Theta_j \rightarrow \infty$.

We partition the matrices

$$A = [B, N], \quad \Theta^{-1} = \begin{bmatrix} \Theta_B^{-1} & 0 \\ 0 & \Theta_N^{-1} \end{bmatrix},$$

where B is an $m \times m$ nonsingular matrix. As done in [8], we permute the columns of A and Θ^{-1} such that $\theta_1^{-1} \leq \theta_2^{-1} \leq \dots \leq \theta_n^{-1}$, and we pick the first m linearly independent columns of A in this order to construct B . The indefinite matrix in (7) can then be rewritten in the following form:

$$K = \begin{bmatrix} -\Theta_B^{-1} & B^T \\ B & -\Theta_N^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}. \quad (8)$$

The preconditioner P is constructed as follows:

$$P = \begin{bmatrix} & B^T \\ B & -\Theta_N^{-1} \end{bmatrix}. \quad (9)$$

P is easily invertible because it is a block-triangular matrix with nonsingular diagonal blocks B , Θ_N^{-1} and B^T .

The PCG method is used to solve the augmented system preconditioned by the block triangular matrix P . A consequence of using PCG method is that the search

direction is computed approximately; this yields a specific inexact interior point method. This causes a major difference to the interior point algorithm, whose convergence is proved under the assumption that the search directions are calculated exactly. In this paper we present the convergence analysis of an infeasible path-following algorithm in which the search directions are computed inexactly. We call this method an inexact infeasible path-following algorithm (IIPF).

The use of inexact Newton methods in interior point methods for LP was investigated in [7, 10–14]. In [10] the convergence of the infeasible interior point algorithm of Kojima, Megiddo, and Mizuno is proved under the assumption that the iterates are bounded. Monteiro and O’Neal [14] propose the convergence analysis of inexact infeasible long-step primal-dual algorithm and give complexity results for this method. In [14] the PCG method is used to solve the normal equations preconditioned with a sparse preconditioner. The proposed preconditioner was inspired by the maximum weight basis algorithm developed in [15]. In [16] an inexact interior point method for semidefinite programming is presented. It allows the linear system to be solved to a low accuracy when the current iterate is far from the solution. In [17] the convergence analysis of inexact infeasible primal-dual path-following algorithm for convex quadratic programming is presented. In these papers the search directions are inexact as the PCG method is used to solve the normal equations. Korzak [18] proves the convergence of the inexact infeasible interior point algorithm of Kojima, Megiddo and Mizuno for LP. This is for search directions which are computed approximately for any iterative solver. This convergence is proven under the assumption that the iterates are bounded. Furthermore, in [19] Zhou and Toh show that the primal-dual inexact infeasible interior point algorithm can find the ϵ -approximate solution of a semidefinite program in $O(n^2 \ln(1/\epsilon))$ iterations. That is also for search directions which are computed approximately for any iterative solver without the need of assuming the boundedness of the iterations. That is because residuals satisfy specific conditions. One of these conditions is dependent on the smallest singular value of the constraint matrix.

In this paper we study the convergence analysis of inexact infeasible path following algorithm for linear programming as the PCG method is used to solve the augmented system preconditioned with block triangular sparse preconditioner. We prove the global convergence and the complexity result for this method without having to assume the boundedness of the iterates. We design a suitable stopping criteria for the PCG method. This plays an important role in the whole convergence of IIPF algorithm. This stopping criteria allows a low accuracy when the current iterate is far from the solution. We state conditions on the forcing term of inexact Newton method in order to prove the convergence of IIPF algorithm.

The approach in this paper can be used in the cases where the augmented system is solved iteratively, providing that the residual of this iterative method has a zero block $r = [r_1, 0]$. So we can carry out the approach in this paper to cases like [20] for example.

In order to prove the convergence of an inexact infeasible interior point method, we should prove first that the PCG method, when applied to an indefinite system, converges. Then, we prove the convergence of the IIPF Algorithm.

The paper is organized as follows. In Sect. 2 we study the behaviour of the residual of the PCG method when applied to an indefinite system (augmented system) precon-

ditioned with (9). In Sect. 3 we compute the residual of the inexact Newton method and choose suitable stopping criteria to the PCG method which makes sense for the convergence of the inexact Newton method. In Sect. 4 we perform the convergence analysis and provide the complexity result for the IIPF Algorithm. In Sect. 5 we summarize numerical results from [8], which demonstrate the behaviour of our approach on some medium scale problems. In Sect. 6 we draw some conclusions.

2 Convergence of the PCG Method

Following the theory developed by Rozložník and Simoncini [20], in [8] we studied the behavior of the preconditioned conjugate gradient method on the indefinite system (7) preconditioned with (9). We can apply PCG method to an indefinite system because the following properties are satisfied. The first property is the preconditioned matrix KP^{-1} is J -symmetric where $J = P^{-1}$, and the second is that the residuals of the PCG method have zero block in the form $r_{PCG}^k = [r_1^k, 0]$ which results from the use of a specific starting point. See [8]. We gave explicit formulae describing the convergence of the error term. The following theorem, which is proved in [8], shows that the convergence of the *error term* is similar to that in the case of symmetric positive definite matrices.

Let e^j and r_{PCG}^j be the error term and the residual term on j -th PCG iteration respectively. The matrix of the augmented system K is indefinite. Therefore, the K -norm is not defined, but the K -inner product $(e^j)^T K e^j$ is always positive (Lemma 4 in [8]). So we allow ourselves to write $\|e^j\|_K = \sqrt{(e^j)^T K e^j}$. Accordingly to the partitioning of K in (8) we will partition the error $e^j = [e_1^j, e_2^j]$, where $e_1^j = [e_B^j, e_N^j]$. Later in this section, we will use the same partitioning for the residual vector $r_{PCG}^j = [r_1^j, r_2^j]$, where $r_1^j = [r_B^j, r_N^j]$.

Theorem 2.1 *Let e^0 be the initial error of PCG. Then,*

$$\begin{aligned} \|e^j\|_K^2 \leq & \min_{\phi \in P_j, \phi(0)=1} \max_{\lambda \in \Lambda(I_m + WW^T)} [\phi(\lambda)]^2 \|e_B^0\|_{\Theta_B^{-1}}^2 \\ & + \min_{\phi \in P_j, \phi(0)=1} \max_{\lambda \in \Lambda(I_{n-m} + W^T W)} [\phi(\lambda)]^2 \|e_N^0\|_{\Theta_N^{-1}}^2, \end{aligned} \quad (10)$$

where P_j is a polynomial of degree j , $\Lambda(G)$ is the set of eigenvalues of the matrix G and $W = \Theta_B^{-1/2} B^{-1} N \Theta_N^{1/2}$. $I_m + WW^T$ and $I_{n-m} + W^T W$ are symmetric positive-definite matrices.

Theorem 2.1 states that the K -norm of the *error* e^j is minimized over the eigenvalues of the symmetric positive definite matrices $I_m + WW^T$ and $I_{n-m} + W^T W$. Consequently, the error term displays asymptotic convergence similar to that observed when PCG is applied to positive definite system.

In the rest of this section we show that the convergence of the *residual* of the PCG method applied to (7) with the preconditioner (9) is similar to the convergence observed in the case of PCG method applied to symmetric positive definite matrix.

2.1 Residual of the PCG Method

The Euclidean norm of the residual is minimized over the eigenvalues of the symmetric positive definite matrix $I_m + WW^T$. The following theorem shows that the residual term displays asymptotic convergence similar to that observed when PCG is applied to positive definite system.

Theorem 2.2 *The residual of the PCG method which is used to solve the augmented system (7) preconditioned by P satisfies*

$$\|r_{PCG}^j\| \leq \min_{\phi \in P_j, \phi(0)=1} \max_{\lambda \in \Lambda(I_m + WW^T)} |\phi(\lambda)| \|r_B^0\|. \quad (11)$$

Proof The residual satisfies

$$r_{PCG}^j = -Ke^j,$$

and the error can be written as

$$e^j = \phi_j(P^{-1}K)e^0.$$

So we can write the residual as

$$r_{PCG}^j = -K\phi_j(P^{-1}K)e^0 = -\phi_j(KP^{-1})Ke^0 = \phi_j(KP^{-1})r_{PCG}^0.$$

Furthermore,

$$KP^{-1}r_{PCG}^0 = \begin{bmatrix} (I + \Theta_B^{-1}B^{-1}N\Theta_N N^T B^{-T})r_B^0 - \Theta_B^{-1}B^{-1}N\Theta_N r_N^0 + \Theta_B^{-1}B^{-1}r_2^0 \\ r_N^0 \\ r_2^0 \end{bmatrix},$$

where $r_{PCG}^j = [r_B^j, r_N^j, r_2^j]$. The initial residual has the form $r_{PCG}^0 = [r_B^0, 0, 0]$ because of using the following starting point:

$$t^{(0)} = \begin{bmatrix} B^{-1}(g - N\Theta_N f_N) \\ \Theta_N f_N \\ 0 \end{bmatrix};$$

see [8], so the previous equation becomes

$$KP^{-1}r_{PCG}^0 = \begin{bmatrix} \Theta_B^{-1}(\Theta_B + B^{-1}N\Theta_N N^T B^{-T})r_B^0 \\ 0 \\ 0 \end{bmatrix}. \quad (12)$$

Let us define

$$C = \Theta_B + B^{-1}N\Theta_N N^T B^{-T}.$$

It is easy to prove that C is a symmetric positive-definite matrix. By repeating a similar argument to one used to drive (12), we obtain

$$r_{PCG}^j = \phi_j(KP^{-1})r_{PCG}^0 = \begin{bmatrix} \phi_j(\Theta_B^{-1}C)r_B^0 \\ 0 \\ 0 \end{bmatrix}, \quad (13)$$

and so

$$\|r_{PCG}^j\| = \|\phi_j(\Theta_B^{-1}C)r_B^0\|. \quad (14)$$

Let us observe that

$$(\Theta_B^{-1}C)^k = \Theta_B^{-1/2}(\Theta_B^{-1/2}C\Theta_B^{-1/2})^k\Theta_B^{1/2} = \Theta_B^{-1/2}(I_m + WW^T)^k\Theta_B^{1/2},$$

where $I_m + WW^T$ is a symmetric positive-definite matrix.

Using these definitions, (14) can be written as

$$\|r_{PCG}^j\| = \|\Theta_B^{-1/2}\phi_j(I_m + WW^T)\Theta_B^{1/2}r_B^0\| = \|\phi_j(I_m + WW^T)\Theta_B^{1/2}r_B^0\|_{\Theta_B^{-1}}.$$

Therefore,

$$\|r_{PCG}^j\| \leq \min_{\phi \in P_j, \phi(0)=1} \max_{\lambda \in \Lambda(I_m + WW^T)} |\phi(\lambda)| \|\Theta_B^{1/2}r_B^0\|_{\Theta_B^{-1}},$$

and the claim is proved after substituting

$$\|\Theta_B^{1/2}r_B^0\|_{\Theta_B^{-1}} = \|r_B^0\|. \quad \square$$

In this section, we proved that the PCG method applied to the indefinite system (7) preconditioned with (9) and initialized with an appropriate starting point, converges similarly to the case of when PCG is applied to positive definite system. In the next section we show that applying PCG to solve (7) with the preconditioner (9) can be analysed using the classical framework of the inexact Newton method (4).

3 Residual of the Inexact Newton Method

Using an iterative method to solve the augmented system (7) produces a specific value of the residual of the inexact Newton method (4). So we shall find the value of the residual r in (4) in order to satisfy (5) and prove the convergence of inexact infeasible path following algorithm.

Solving (7) approximately gives

$$\begin{bmatrix} -\Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \quad (15)$$

where $r_1 = [r_B, r_N]$.

That gives the following equations:

$$-X^{-1}S\Delta x + A^T\Delta y = f + r_1 = c - A^Ty - \sigma\mu X^{-1}e + r_1, \quad (16)$$

$$A\Delta x = g + r_2 = b - Ax + r_2. \quad (17)$$

Then we find Δs by substituting Δx in (6). However, we can shift the residual from (16) to (6) by assuming there is a residual h while computing Δs . Then, (6) is replaced by

$$\Delta s = -X^{-1}S\Delta x - s + \sigma\mu X^{-1}e + h,$$

which we can rewrite as

$$-X^{-1}S\Delta x = \Delta s + s - \sigma\mu X^{-1}e - h.$$

Substituting it in (16) gives

$$A^T\Delta y + \Delta s = c - A^Ty - s + h + r_1.$$

To satisfy the second equation of (3), we choose $h = -r_1$. This gives

$$A^T\Delta y + \Delta s = c - A^Ty - s, \quad (18)$$

and

$$\Delta s = -X^{-1}S\Delta x - s + \sigma\mu X^{-1}e - r_1,$$

which implies

$$S\Delta x + X\Delta s = -XSe + \sigma\mu e - Xr_1. \quad (19)$$

Equations (17), (18) and (19) give

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix} + \begin{bmatrix} r_2 \\ 0 \\ -Xr_1 \end{bmatrix},$$

where

$$\xi_p = b - Ax, \quad \xi_d = c - A^Ty - s, \quad \xi_\mu = -XSe + \sigma\mu e, \quad \sigma \in (0, 1).$$

In the setting in which we apply the PCG method to solve (7) preconditioned with (9) we have $r_2 = 0$ and $r_1 = [r_B, 0]$, see (13) in the proof of Theorem 2.2. Therefore, the inexact Newton method residual r is

$$r = \begin{bmatrix} 0 \\ 0 \\ -Xr_1 \end{bmatrix},$$

with

$$Xr_1 = \begin{bmatrix} X_B r_B \\ X_N r_N \end{bmatrix} = \begin{bmatrix} X_B r_B \\ 0 \end{bmatrix}.$$

Shifting the residual from (16) to (6) is an essential step to prove the convergence of the IIPF algorithm. It results in moving the residual from the second row to the last row of the inexact Newton system, which makes the proof of the convergence of the IIPF algorithm much easier, as we will see in Sect. 4.

The issue of choosing the stopping criteria of inexact Newton method has been discussed in many papers. See for example [10, 11, 16]. The inexact Newton method residual in these papers is chosen such that

$$\|r^k\| \leq \eta_k \mu_k.$$

Let the residual be $r = [r_p, r_d, r_\mu]$. In our case, $r_p = r_d = 0$; we will stop the PCG algorithm when

$$\|r_\mu^k\|_\infty \leq \eta_k \mu_k.$$

As $r_\mu^k = -X^k r_1^k$ and $r_1 = [r_B, 0]$, the stopping criterion becomes

$$\|X_B^k r_B^k\|_\infty \leq \eta_k \mu_k. \quad (20)$$

We terminate the PCG algorithm when the stopping criterion (20) is satisfied. This stopping criteria allows a low accuracy when the current iterate is far from the solution. In the later iterations the accuracy increases because the average complementarity gap μ reduces from one iteration to another.

4 Convergence of the IIPF Algorithm

In this section, we carry out the proof of the convergence of the IIPF algorithm and derive a complexity result. In the previous section we used the shifting residual strategy, which makes the proof of the convergence of this inexact algorithm similar to that of the exact case.

This section is organised as follows. First we describe the IIPF algorithm. Then in Lemmas 4.1 and 4.2 we derive useful bounds on the iterates. In Lemma 4.3 and Theorem 4.1 we prove that there is a steplength α such that the new iteration generated by IIPF algorithm belongs to the neighbourhood $\mathcal{N}_{-\infty}(\gamma, \beta)$ and the average complementarity gap decreases. In order to prove that we supply conditions on the forcing term η_k . In Theorem 4.2 we show that the sequence $\{\mu_k\}$ converges Q-linearly to zero and the normal residual sequence $\{\|(\xi_p^k, \xi_d^k)\|\}$ converges R-linearly to zero. Finally in Theorem 4.3, we provide the complexity result for this algorithm.

Definition 4.1 The central path neighborhood $\mathcal{N}_{-\infty}(\gamma, \beta)$ is defined by

$$\begin{aligned} \mathcal{N}_{-\infty}(\gamma, \beta) = \{ & (x, y, s) : \|(\xi_p, \xi_d)\|/\mu \leq \beta \|(\xi_p^0, \xi_d^0)\|/\mu_0, \\ & (x, s) > 0, \ x_i s_i \geq \gamma \mu, \ i = 1, 2, \dots, n\}, \end{aligned} \quad (21)$$

where $\gamma \in (0, 1)$ and $\beta \geq 1$ [2].

4.1 Inexact Infeasible Path-Following Algorithm

Step 1. Given $\gamma, \beta, \sigma_{\min}, \sigma_{\max}$ with $\gamma \in (0, 1)$, $\beta \geq 1$, $0 < \sigma_{\min} < \sigma_{\max} < 0.5$, and $0 < \eta_{\min} < \eta_{\max} < 1$; choose (x^0, y^0, s^0) with $(x^0, s^0) > 0$;

Step 2. For $k = 0, 1, 2, \dots$

- choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$ and $\eta_k \in [\eta_{\min}, \eta_{\max}]$; solve

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} \xi_p^k \\ \xi_d^k \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -X^k r_1^k \end{bmatrix}, \quad (22)$$

such that $r_N^k = 0$ and

$$\|X_B^k r_B^k\|_{\infty} \leq \eta_k \mu_k, \quad (23)$$

- choose α_k as the largest value of α in $[0, 1]$ such that

$$(x^k(\alpha), y^k(\alpha), s^k(\alpha)) \in \mathcal{N}_{-\infty}(\gamma, \beta) \quad (24)$$

and the following Armijo condition holds:

$$\mu_k(\alpha) \leq (1 - 0.01\alpha)\mu_k; \quad (25)$$

- set $(x^{k+1}, y^{k+1}, s^{k+1}) = (x^k(\alpha_k), y^k(\alpha_k), s^k(\alpha_k))$;
- stop when $\mu_k < \epsilon$, for a small positive constant ϵ .

In this section, we will follow the convergence analysis of the infeasible path-following algorithm proposed originally by Zhang [3]. However, we will follow the proof techniques proposed in Wright's book [2].

First, let us introduce the quantity

$$v_k = \prod_{j=0}^{k-1} (1 - \alpha_j), \quad v_0 = 1.$$

Using similar argument to that applied in [2], we observe that $\xi_p^k = v_k \xi_p^0$ and $\xi_d^k = v_k \xi_d^0$.

Let (x^*, y^*, s^*) be any primal-dual solution.

Lemma 4.1 Assume that $(x^k, y^k, s^k) \in \mathcal{N}_{-\infty}(\gamma, \beta)$, $(\Delta x^k, \Delta y^k, \Delta s^k)$ satisfies (22) and (23) for all $k \geq 0$, and $\mu_k \leq (1 - 0.01\alpha_{k-1})\mu_{k-1}$ for all $k \geq 1$. Then, there is a positive constant C_1 such that, for all $k \geq 0$,

$$v_k \|(x^k, s^k)\| \leq C_1 \mu_k, \quad (26)$$

where C_1 is given as

$$C_1 = \zeta^{-1}(n\beta + n + \beta) \|(x^0, s^0)\|_{\infty} \|(x^*, s^*)\|_1 / \mu_0,$$

where

$$\zeta = \min_{i=1,\dots,n} \min(x_i^0, s_i^0).$$

The proof of this lemma is similar to the proof of Lemma 6.3 in [2].

For simplicity, we omit the iteration index k in all the lemmas' proofs.

Lemma 4.2 Assume that $(x^k, y^k, s^k) \in \mathcal{N}_{-\infty}(\gamma, \beta)$, $(\Delta x^k, \Delta y^k, \Delta s^k)$ satisfies (22) and (23) for all $k \geq 0$, and $\mu_k \leq (1 - 0.01\alpha_{k-1})\mu_{k-1}$ for all $k \geq 1$. Then, there is a positive constant C_2 such that

$$\|D^{-1}\Delta x^k\| \leq C_2\mu_k^{1/2}, \quad (27)$$

$$\|D\Delta s^k\| \leq C_2\mu_k^{1/2}, \quad (28)$$

where $D = X^{1/2}S^{-1/2}$. For all $k \geq 0$.

Proof The proof of this lemma is similar to the proof of Lemma 6.5 in [2]. However, we need to find a bound on $\|-XSe + \sigma\mu e - Xr_1\|$ instead of $\|-XSe + \sigma\mu e\|$ in [2, Lemma 6.5].

$$\|-XSe + \sigma\mu e - Xr_1\| \leq \|-XSe + \sigma\mu e\| + \|Xr_1\|.$$

From [2, Lemma 6.5], we have

$$\|-XSe + \sigma\mu e\| \leq n\mu.$$

We first observe that

$$\|-XSe + \sigma\mu e - Xr_1\| \leq n\mu + \sqrt{n}\|X_B r_B\|_{\infty} \leq n\mu + \sqrt{n}\eta\mu \leq n\mu + \sqrt{n}\eta_{\max}\mu.$$

The rest of the proof follows [2, Lemma 6.5] with n replaced by $(n + \sqrt{n}\eta_{\max})$. So in our case the positive constant C_2 is given by the following formula;

$$C_2 = \gamma^{-1/2}(n + \sqrt{n}\eta_{\max}) + 2C_1\gamma^{-1/2} \max(\|x^0 - x^*\|, \|s^0 - s^*\|). \quad \square$$

Lemma 4.3 Assume that $(x^k, y^k, s^k) \in \mathcal{N}_{-\infty}(\gamma, \beta)$, $(\Delta x^k, \Delta y^k, \Delta s^k)$ satisfies (22) and (23) for all $k \geq 0$, and $\mu_k \leq (1 - 0.01\alpha_{k-1})\mu_{k-1}$ for all $k \geq 1$. Then, there is a value $\tilde{\alpha} \in (0, 1)$ such that the following three conditions are satisfied for all $\alpha \in [0, \tilde{\alpha}]$ for all $k \geq 0$:

$$(x^k + \alpha\Delta x^k)^T (s^k + \alpha\Delta s^k) \geq (1 - \alpha)(x^k)^T s^k, \quad (29)$$

$$(x_i^k + \alpha\Delta x_i^k)(s_i^k + \alpha\Delta s_i^k) \geq \frac{\gamma}{n}(x^k + \alpha\Delta x^k)^T (s^k + \alpha\Delta s^k), \quad (30)$$

$$(x^k + \alpha\Delta x^k)^T (s^k + \alpha\Delta s^k) \leq (1 - 0.01\alpha)(x^k)^T s^k. \quad (31)$$

Proof The proof of this lemma is similar to the proof of Lemma 6.7 in [2].

Let us consider (29). From [2, Lemma 6.7], we have

$$(x + \alpha \Delta x)^T (s + \alpha \Delta s) - (1 - \alpha)x^T s \geq (n\alpha\sigma - \alpha^2 C_3)\mu,$$

where $C_3 = C_2^2$.

However, in our case we should add the term $-\alpha x_B^T r_B$. Consequently,

$$(x + \alpha \Delta x)^T (s + \alpha \Delta s) - (1 - \alpha)x^T s \geq (n\alpha\sigma - \alpha^2 C_3)\mu - \alpha x_B^T r_B.$$

From (23), we have

$$|x_B^T r_B| \leq n \|X_B r_B\|_\infty \leq n\eta\mu.$$

This leads to

$$(x + \alpha \Delta x)^T (s + \alpha \Delta s) - (1 - \alpha)x^T s \geq n\alpha\sigma\mu - n\alpha\eta\mu - \alpha^2 C_3\mu.$$

Therefore, the condition (29) holds for all $\alpha \in [0, \alpha_1]$, where α_1 is given by

$$\alpha_1 = \frac{n(\sigma - \eta)}{C_3}, \quad (32)$$

and we choose $\eta < \sigma - \varepsilon_1$ to guarantee α_1 to be strictly positive, where ε_1 is a constant strictly greater than zero.

Now, let us consider (30). Also from [2, Lemma 6.7], we have

$$(x_i + \alpha \Delta x_i)(s_i + \alpha \Delta s_i) - \frac{\gamma}{n}(x + \alpha \Delta x)^T (s + \alpha \Delta s) \geq \alpha\sigma(1 - \gamma)\mu - 2\alpha^2 C_3\mu.$$

However, in our case, we should add the term $\alpha(-x_i r_{1,i} + \frac{\gamma}{n}x_B^T r_B)$. That leads to

$$\begin{aligned} & (x_i + \alpha \Delta x_i)(s_i + \alpha \Delta s_i) - \frac{\gamma}{n}(x + \alpha \Delta x)^T (s + \alpha \Delta s) \\ & \geq \alpha\sigma(1 - \gamma)\mu - 2\alpha^2 C_3\mu + \alpha \left(-x_i r_{1,i} + \frac{\gamma}{n}x_B^T r_B \right). \end{aligned}$$

Furthermore, we have

$$-x_i r_{1,i} + \frac{\gamma}{n}x_B^T r_B \geq -\|X_B r_B\|_\infty - \frac{\gamma}{n}|x_B^T r_B| \geq \eta\mu - \eta\gamma\mu.$$

This gives

$$\begin{aligned} & (x_i + \alpha \Delta x_i)(s_i + \alpha \Delta s_i) - \frac{\gamma}{n}(x + \alpha \Delta x)^T (s + \alpha \Delta s) \\ & \geq \alpha((1 - \gamma)\sigma - \eta(1 + \gamma))\mu - \alpha^2 \left(1 + \frac{\gamma}{n} \right) C_3\mu. \end{aligned}$$

Condition (30) holds for all $\alpha \in [0, \alpha_2]$, where α_2 is given by

$$\alpha_2 = \frac{\sigma(1-\gamma) - (1+\gamma)\eta}{(1+\frac{\gamma}{n})C_3}. \quad (33)$$

We choose $\eta < \frac{\sigma(1-\gamma)}{(1+\gamma)} - \varepsilon_2$ to guarantee α_2 to be strictly positive, where ε_2 is a constant strictly greater than zero.

Finally, let us consider condition (31). From [2, Lemma 6.7], we have

$$\frac{1}{n}[(x + \alpha \Delta x)^T(s + \alpha \Delta s) - (1 - 0.01\alpha)x^T s] \leq -0.99\alpha\mu + \alpha\sigma\mu + \frac{\alpha^2}{n}C_3\mu.$$

In our case, we should add the following term $-\alpha x_B^T r_B$. That gives

$$\begin{aligned} & \frac{1}{n}[(x + \alpha \Delta x)^T(s + \alpha \Delta s) - (1 - 0.01\alpha)x^T s] \\ & \leq -0.99\alpha\mu + \alpha\sigma\mu + \frac{\alpha^2}{n}C_3\mu - \alpha x_B^T r_B \\ & \leq -0.99\alpha\mu + \alpha\sigma\mu + \alpha\eta\mu + \frac{\alpha^2}{n}C_3\mu. \end{aligned}$$

We can conclude that condition (31) holds for all $\alpha \in [0, \alpha_3]$, where α_3 is given by

$$\alpha_3 = \frac{n(0.99 - \sigma - \eta)}{C_3}. \quad (34)$$

We choose η and σ such that $\eta + \sigma < 0.99 - \varepsilon_3$ to guarantee α_3 to be strictly positive, where ε_3 is a constant strictly greater than zero.

Combining the bounds (32), (33) and (34), we conclude that conditions (29), (30) and (31) hold for $\alpha \in [0, \bar{\alpha}]$, where

$$\bar{\alpha} = \min \left\{ 1, \frac{n(\sigma - \eta)}{C_3}, \frac{\sigma(1-\gamma) - (1+\gamma)\eta}{(1+\frac{\gamma}{n})C_3}, \frac{n(0.99 - \sigma - \eta)}{C_3} \right\}. \quad (35)$$

□

We introduce the constants ε_1 , ε_2 and ε_3 to guarantee that the limit of the step length $\bar{\alpha}$ is strictly greater than zero and to make it flexible to choose the parameters η_k and σ_k .

Note that, if $\eta < \frac{\sigma(1-\gamma)}{(1+\gamma)}$, then $\eta < \sigma$ because $\frac{(1-\gamma)}{(1+\gamma)} < 1$ for any $\gamma \in (0, 1)$.

From the proof of Lemma 4.3, we deduce the forcing term η_k should be chosen such that the two conditions

$$\eta_k < \frac{\sigma_k(1-\gamma)}{(1+\gamma)} - \varepsilon_2 \quad \text{and} \quad \eta_k + \sigma_k < 0.99 - \varepsilon_3$$

are satisfied. Under these assumption the following theorem guarantees that there is a step length α such that the new point belongs to the neighbourhood $\mathcal{N}_{-\infty}(\gamma, \beta)$ and its average complementarity gap decreases according to condition (25).

The proofs of the following two theorems directly follow Wright's [2] analysis.

Theorem 4.1 Assume that $\eta_k < \frac{\sigma_k(1-\gamma)}{(1+\gamma)} - \varepsilon_2$, $\eta_k + \sigma_k < 0.99 - \varepsilon_3$ for $\varepsilon_2, \varepsilon_3 > 0$, $(x^k, y^k, s^k) \in \mathcal{N}_{-\infty}(\gamma, \beta)$ and $(\Delta x^k, \Delta y^k, \Delta s^k)$ satisfies (22) and (23) for all $k \geq 0$, $\mu_k \leq (1 - 0.01\alpha_{k-1})\mu_{k-1}$ for all $k \geq 1$. Then, $(x^k(\alpha), y^k(\alpha), s^k(\alpha)) \in \mathcal{N}_{-\infty}(\gamma, \beta)$ and $\mu_k(\alpha) \leq (1 - 0.01\alpha)\mu_k$ for all $\alpha \in [0, \bar{\alpha}]$, where $\bar{\alpha}$ is given by (35).

Theorem 4.2 The sequence $\{\mu_k\}$ generated by the IIPF Algorithm converges Q -linearly to zero, and the sequence of residual norms $\{\|(\xi_p^k, \xi_d^k)\|\}$ converges R -linearly to zero.

Theorem 4.3 Let $\epsilon > 0$ and the starting point $(x^0, y^0, s^0) \in \mathcal{N}_{-\infty}(\gamma, \beta)$ in the Algorithm IIPF be given. Then, there is an index K with

$$K = O(n^2 |\log \epsilon|)$$

such that the iterates $\{(x^k, y^k, s^k)\}$ generated by IIPF Algorithm satisfy

$$\mu_k \leq \epsilon, \quad \text{for all } k \geq K.$$

Proof If the conditions of Theorem 4.1 are satisfied, then the conditions (24) and (25) are satisfied for all $\alpha \in [0, \bar{\alpha}]$ for all $k \geq 0$. By Lemma 4.3, the quantity $\bar{\alpha}$ satisfies

$$\bar{\alpha} \geq \min \left\{ 1, \frac{n(\sigma - \eta)}{C_3}, \frac{\sigma(1 - \gamma) - (1 + \gamma)\eta}{(1 + \frac{\gamma}{n})C_3}, \frac{n(0.99 - \sigma - \eta)}{C_3} \right\}.$$

Furthermore, from Lemmas 4.1, 4.2 and 4.3 we have $C_3 = O(n^2)$; therefore,

$$\bar{\alpha} \geq \frac{\delta}{n^2}$$

for some positive scalar δ independent of n . That implies

$$\mu_{k+1} \leq (1 - 0.01\bar{\alpha})\mu_k \leq \left(1 - \frac{0.01\delta}{n^2}\right)\mu_k, \quad \text{for } k \geq 0.$$

The complexity result is an immediate consequence of Theorem 3.2 of [2]. □

5 Numerical Results

The PCG method preconditioned by the block triangular matrix P , which is proposed in this paper, has been implemented in the context of HOPDM [21]. See [8] for more details. This iterative approach works best at the final iterations of the IPM, when the direct approach starts to struggle in solving the normal equations because of the extremely bad ill-conditioning of this system. Therefore, in the initial iterations of the interior point method the normal equations are solved using the direct approach

Table 1 Comparing the mix approach and the direct approach

Problem	Dimensions		Memory saving	Total CPU time saving	Average number of PCG iterations
	m	n			
aircraft	3754	7517	99.3%	24.8%	9
chr12a	947	1662	92.6%	4.61%	20
chr12b	947	1662	94.9%	11.9%	29
chr12c	947	1662	92.3%	0.78%	26
chr15b	1814	3270	95.6%	5.30%	38
chr15c	1814	3270	95.5%	3.01%	32
chr18a	3095	5679	95.5%	11.3%	37
chr18b	3095	5679	96.3%	8.72%	56
chr20a	4219	7810	95.7%	23.1%	56
chr20b	4219	7810	92.9%	12.5%	63
chr22a	5587	10417	97.5%	3.36%	49
chr22b	5587	10417	97.2%	44.5%	42
chr25a	8148	15325	98.1%	11.3%	50
fit1p	628	1677	97.5%	42.2%	3
fit2p	3001	13525	99.2%	63.7%	15
fome10	6071	12230	92.9%	56.0%	243
fome11	14695	24460	92.6%	65.2%	205
fome12	24285	48920	93.1%	63.3%	210
pds-06	9882	28655	96.2%	57.8%	53
pds-10	16559	48763	97.7%	47.8%	60
pds-20	33875	105728	97.7%	61.6%	78
route	20894	23923	99.6%	9.20%	60
scr10	689	1540	89.0%	18.4%	19
scr12	1151	2784	93.8%	18.8%	45
scr15	2234	6210	38.1%	18.8%	61
scr20	5079	15980	93.2%	15.8%	181

by forming the Cholesky factorisation LDL^T for the normal equations matrix. As the interior point method approaches optimality, we switch to the iterative solver. This mixed approach can be very effective in some cases, and it is also an important option for some classes of problems. See [8].

The results for problems which benefit from the use of this mixed approach are reported in [8]. The memory required to store the LU factorisation for the basis matrix B (iterative solver) compared with the Cholesky factorisation to the normal equations (direct solver), the results of comparing our mixed approach against the pure direct approach and the number of PCG iterations during IPM iterations are presented in [8].

We summarise some of these numerical results in Table 1. In this table, we report the problem dimensions: m and n denote the number of rows and columns in the constraint matrix A . In the next two columns, we present the saving in the storage

memory and the total CPU time, when the mixed approach is used instead of the pure direct approach. The iterative method is storage-efficient, requiring one or two orders of magnitude less storage than the Cholesky factorisation. In pds20 problem for instance, the Cholesky factorisation has 1626987 nonzeros, while LU factorisation only has 37123. So the memory saving reach 97.7%. The total CPU time in seconds is calculated for two approaches, the pure direct approach and the mixed approach. Many problems have responded very well when the mixed approach is used. The final column of this table illustrates the average number of the PCG iterations. The number of PCG iterations during IPM gives an idea about the behaviour of the PCG method.

6 Conclusions

In this paper we have considered the convergence analysis of the inexact infeasible path-following algorithm, where the augmented system is solved iteratively. We have analysed the behaviour of the residual term in the PCG method which is used to solve the augmented system (indefinite system). This analysis reveals that the residual converges to zero and, asymptotically, behaves in a similar way to the classical case when PCG is applied to positive definite system. We have chosen a suitable stopping criteria of the PCG method and have provided a condition on the forcing term. Furthermore, we have proved the global convergence of the IIPF algorithm and have provided a complexity result for this method. The technique to control accuracy in the inexact Newton method proposed and analysed in this paper has been implemented in HOPDM [8].

Acknowledgements We are grateful to the Anonymous Referee and Associate Editor for their comments which helped us to improve the paper.

References

1. Andersen, E.D., Gondzio, J., Mészáros, C., Xu, X.: Implementation of interior point methods for large scale linear programming. In: Terlaky, T. (ed.) *Interior Point Methods in Mathematical Programming*, pp. 189–252. Kluwer Academic, Amsterdam (1996)
2. Wright, S.J.: *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia (1997)
3. Zhang, Y.: On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem. *SIAM J. Optim.* **4**, 208–227 (1994)
4. Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact Newton methods. *SIAM J. Numer. Anal.* **19**, 400–408 (1982)
5. Kelley, C.T.: *Iterative Methods for Linear and Nonlinear Equations*. *Frontiers in Applied Mathematics*, vol. 16. SIAM, Philadelphia (1995)
6. Bocanegra, S., Campos, F., Oliveira, A.: Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods. *Comput. Optim. Appl.* **36**, 149–164 (2007)
7. Chai, J.S., Toh, K.C.: Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming. *Comput. Optim. Appl.* **36**, 221–247 (2007)
8. Al-Jeiroudi, G., Gondzio, J., Hall, J.: Preconditioning indefinite systems in interior point methods for large scale linear optimization. *Optim. Methods Softw.* **23**, 345–363 (2008)
9. Oliveira, A.R.L., Sorensen, D.C.: A new class of preconditioners for large-scale linear systems from interior point methods for linear programming. *Linear Algebra Appl.* **394**, 1–24 (2005)

10. Baryamureeba, V., Steihaug, T.: On the convergence of an inexact primal-dual interior point method for linear programming. Technical report, University of Bergen (2000)
11. Bellavia, S.: An inexact interior point method. *J. Optim. Theory Appl.* **96**, 109–121 (1998)
12. Freund, R.W., Jarre, F., Mizuno, S.: Convergence of a class of inexact interior-point algorithms for linear programs. *Math. Oper. Res.* **24**, 105–122 (1999)
13. Mizuno, S., Jarre, F.: Global and polynomial-time convergence of an infeasible-interior-point algorithm using inexact computation. *Math. Program.* **84**, 105–122 (1999)
14. Monteiro, R.D.S., O’Neal, J.W.: Convergence analysis of long-step primal-dual infeasible interior point LP algorithm based on iterative linear solvers. Georgia Institute of Technology (2003)
15. Resende, M.G.C., Veiga, G.: An implementation of the dual affine scaling algorithm for minimum cost flow on bipartite uncapacitated networks. *SIAM J. Optim.* **3**, 516–537 (1993)
16. Bellavia, S., Pieraccini, S.: Convergence analysis of an inexact infeasible interior point method for semidefinite programming. *Comput. Optim. Appl.* **29**, 289–313 (2004)
17. Lu, Z., Monteiro, R.D.S., O’Neal, J.W.: An iterative solver-based infeasible primal-dual path-following algorithm for convex QP. *SIAM J. Optim.* **17**, 287–310 (2006)
18. Korzák, J.: Convergence analysis of inexact infeasible-interior-point-algorithm for solving linear programming problems. *SIAM J. Optim.* **11**, 133–148 (2000)
19. Zhou, G., Toh, K.C.: Polynomiality of an inexact infeasible interior point algorithm for semidefinite programming. *Math. Program.* **99**, 261–282 (2004)
20. Rozlozník, M., Simoncini, V.: Krylov subspace methods for saddle point problems with indefinite preconditioning. *SIAM J. Matrix Anal. Appl.* **24**, 368–391 (2002)
21. Gondzio, J.: HOPDM (version 2.12)—A fast LP solver based on a primal-dual interior point method. *Eur. J. Oper. Res.* **85**, 221–225 (1995)